

DDM Generation

This section covers the following topics:

- Natural Data Definition Module - DDM
 - SQL Services
-

Natural Data Definition Module - DDM

To enable Natural to access an SQL/DS table, a Natural DDM of the table must be generated. This is done either with Predict (see the relevant Predict documentation for details) or with the Natural utility SYSDDM.

If you do not have Predict installed, use the SYSDDM function "SQL Services" to generate Natural DDMs from SQL/DS tables. This function is invoked from the main menu of SYSDDM and is described below.

SQL Services

If you are using SYSDDM SQL Services in a CICS environment, you should have specified CONVERS=ON in the NDBPARM module.

The SQL Services offer a number of functions related to SQL/DS tables. If you enter function code "B" for "SQL Services" on the main menu of the SYSDDM utility, a menu is displayed, which offers you the following functions:

- Select SQL Table from a List
- Generate DDM from an SQL Table
- List Columns of an SQL Table

The input field "SQL system" contains the name of the SQL system actually in use. If Natural for SQL/DS is the only SQL interface installed, the field is write-protected and contains "SQL/DS"; if not, enter "SQL/DS".

The individual functions are described below.

Select SQL Table from a List

This function is used to select an SQL/DS table from a list for further processing.

To invoke the function, enter function code "S" on the SQL Services Menu. If you enter the function code only, you obtain a list of all tables defined to the SQL/DS catalog.

If you do not want a list of all tables but would like only a certain range of tables to be listed, you can, in addition to the function code, specify a start value in the Table Name and/or Creator fields. You can also use asterisk notation (*) for the start value.

When you invoke the function, the "Select SQL Table from a List" screen is invoked displaying a list of all SQL/DS tables requested.

On the list you can mark an SQL/DS table with either "G" for "Generate DDM from an SQL Table" or "L" for "List Columns of an SQL Table". Then the corresponding function is invoked for the marked table.

Generate DDM from an SQL Table

This function is used to generate a Natural DDM from an SQL/DS table, based on the definitions in the SQL/DS catalog.

To invoke the function, enter function code "G" on the SQL Services Menu along with the name and creator of the table for which you wish a DDM to be generated. If you do not know the table name/creator, you can use the function "Select SQL Table from a List" to choose the table you want.

If you do not want the creator of the table to be part of the DDM name, enter a "N" in the field "DDM Name with Creator" when you invoke the Generate function (default is "Y").

Note:

Since the specification of any special characters as part of a field or DDM name does not comply with Natural naming conventions, any special characters allowed within SQL/DS must be avoided.
SQL/DS delimited identifiers must be avoided, too.

If you wish to generate a DDM for a table for which a DDM already exists and you want the existing one to be replaced by the newly generated one, enter a "Y" in the Replace field when you invoke the Generate function.

By default, Replace is set to "N" to prevent an existing DDM from being replaced accidentally. If Replace is "N", you cannot generate another DDM for a table for which a DDM has already been generated.

DBID/FNR Assignment

When the "Generate DDM from an SQL Table" function is invoked for a table for which a DDM is to be generated for the first time, the DBID/FNR Assignment screen is displayed. If a DDM is to be generated for a table for which a DDM already exists, the existing DBID and FNR are used and the DBID/FNR Assignment screen is suppressed.

On the DBID/FNR Assignment screen, enter one of the database IDs (DBIDs) chosen at Natural installation time and the file number (FNR) to be assigned to the SQL/DS table. Natural requires these specifications for identification purposes only.

The range of DBIDs which are reserved for SQL/DS tables is specified in the NTDB macro of the Natural parameter module (see the relevant section in the Natural Parameter Reference documentation) in combination with the NDBID macro of the parameter module NDBPARAM. Any DBID not within this range is not accepted. The FNR can be any valid number between 1 and 255.

After a valid DBID and FNR have been assigned, a DDM is automatically generated from the specified table.

Long Field Redefinition

The maximum field length supported by Natural is 253 bytes. If an SQL/DS table contains a column which is longer than 253 bytes, this column has to be redefined as a one-dimensional array; otherwise the column is truncated and only the first 253 bytes are considered.

When redefined as an array, this array is represented in the DDM as a multiple-value field. Arrays are defined on the Long Field Redefinition screen, which is automatically invoked for each column over 253 bytes in length.

On the Long Field Redefinition screen you specify the element length of the array, which means the length of the occurrences. The number of occurrences depends on the length you specify.

If, for example, an SQL/DS column has a length of 2000 bytes, you can specify an array element length of 200 bytes, and you receive a multiple-value field with 10 occurrences, each occurrence with a length of 200 bytes.

Since redefined long fields are no multiple-value fields in the sense of Natural, the Natural C* notation cannot be applied to those fields.

When such a redefined long field is defined in a Natural view for being referenced by Natural SQL statements (that is, by host variables which represent multiple-value fields), both when defined and when referenced, the specified range of occurrences (index range) must always start with occurrence 1. If not, a Natural syntax error is returned.

Example:

```
UPDATE table SET varchar = #arr(*)
SELECT ... INTO #arr(1:5)
```

Note:

When such a redefined long field is updated with the Natural DML UPDATE statement, care must be taken to update each occurrence appropriately.

Length Indicator for Variable Length Fields - VARCHAR, LONG VARCHAR, VARGRAPHIC, LONG VARGRAPHIC

For each variable length column, an additional length indicator field (format/length I2) is generated in the DDM. The length is always measured in number of characters, not in bytes. To obtain the number of bytes of a VARGRAPHIC or LONG VARGRAPHIC field, the length must be multiplied by 2.

The name of a length indicator field begins with "L@" followed by the name of the corresponding field. The value of the length indicator field can be checked or updated by a Natural program.

If the length indicator field is not part of the Natural view and if the corresponding field is a redefined long field, the length of this field with UPDATE and STORE operations is calculated without trailing blanks.

Null Values

With Natural, it is possible to distinguish between a null value and the actual value "0" (or "blank") in an SQL/DS column.

When a DDM is generated from the SQL/DS catalog, an additional null indicator field is generated for each column which can be NULL; that is, which has neither "NOT NULL" nor "NOT NULL WITH DEFAULT" specified.

The name of the null indicator field begins with "N@" followed by the name of the corresponding field.

When the column is read from the database, the corresponding indicator field contains either "0" (if the column contains a value, including the value "0" or "blank") or "-1" (if the column contains no value).

Example:

The column "NULLCOL CHAR(6)" in an SQL/DS table definition would result in the following DDM fields:

```
NULLCOL      A      6.0
N@NULLCOL    I      2.0
```

When the field NULLCOL is read from the database, the additional field N@NULLCOL contains:

- "0" if NULLCOL contains a value (including the values "0" and "blank");
- "-1" if NULLCOL contains no value.

A null value can be stored in a database field by providing "-1" as input for the corresponding null indicator field.

Note:

If a column is NULL, an implicit RESET is performed on the corresponding Natural field.

List Columns of an SQL Table

This function lists all columns of a specific SQL/DS table.

To invoke this function, enter function code "L" on the SQL Services Menu along with the name and creator of the table whose columns you wish to be listed.

The List Columns screen for this table is invoked, which lists all columns of the specified table and displays the following information for each column:

Variable	Content
Name	The SQL/DS name of the column.
Type	The column type.
Length	The length (or precision if Type is DECIMAL) of the column as defined in the SQL/DS catalog.
Scale	The decimal scale of the column (only applicable if Type is DECIMAL).
Update	Y The column can be updated. N The column cannot be updated.
Nulls	Y The column can contain null values. N The column cannot contain null values.
Not	A column which is of a scale length or type not supported by Natural is marked with an asterisk (*). For such a column, a view field cannot be generated. The maximum scale length supported is 7 bytes. Types supported are: CHAR, VARCHAR, LONG VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, DECIMAL, INTEGER, SMALLINT, DATE, TIME, TIMESTAMP and FLOAT.

The data types DATE, TIME, TIMESTAMP and FLOAT are converted into numeric or alphanumeric fields of various lengths: DATE is converted into A10, TIME into A8, TIMESTAMP into A26 and FLOAT into F8.

For SQL/DS, Natural provides an SQL/DS TIMESTAMP column as an alphanumeric field (A26) of the format "YYYY-MM-DD-HH.SS.MMMMMM".

Since Natural does not yet support computations with such fields, a Natural subprogram called NDBSTMP is provided to enable this kind of functionality.